Guide to classes and functions developed

Course Project EE559: APS Failure at Scania Trucks

Name: Royston Marian Mascarenhas
USC ID: 8286328166
USC email: rmascare@usc.edu
Spring 2019

I have developed three classes with self written functions.

Functions used reside in three classes

1. Preprocess class
2. Transform class
3. classifier class

Each class has its own py file.

One other standalone function exists : validation_metrics. It is used during model selection. It is a combination of the metric functions in the classifier class.

1. Preprocess class:

Primarily for preprocessing data, compensation for unbalanced data included

Member variables:
data: the dataframe containing the data
n: number of data points
nf: number of features
tr_labels: determined or read through another function. Stores labels.
missflag = if True, read data with missing values replaced by NaN
extflag = if True, read data without missing values

Function	Parameter	Returns	Function
label_splitter	column corresponding to labels, splitflag (split label column from data if True), labels(initialize object's labels variable if splitflag is False)	None	Initialize labels
label_encode	None	None	Label encode labels
feature scraper	trust_threshold (higher the trust threshold, lesser the reduction of	None, return column indexes of discarded	Discard features based on number of missing values. Number of missing values tolerated

	features), trailflag (interactive reduction), retflag	features if retflag = True	is defined by trust threshold.
impute_means_fit_trans	none	none	impute column with mean of feature vector
impute_means_transform	dataframe	dataframe	Impute with means determined by impute_means_fit_trans
impute_classmeans	deprecated	deprecated	deprecated
feature_std	None	None	Standardization fit and transform
feature_std_transform	data	data	Standardization transform based on fit in feature_std
custom_std	doflag (standardize if True)	None	Extract means and standard deviation of feature vectors
resample_smote	None	None	Generate synthetic samples based on SMOTE. Oversample the minority class.

2. Transform class:

For interactive dimensionality reduction

Member variables
data: training data
labels: training labels
tata: test data
tabels : test labels

Function	Parameter	Returns	Function
perfPCA	nc(number of components required), ncflag(if True, do PCA based on nc, if not do PCA based on variance validation), lowvar and upvar are variance thresholds, thresh is the percentage of original number of features that must be preserved.	Transformed data	Perform PCA by either using number of components nc or by obtaining a tradeoff between the required threshold and the specified variance.

pca_transform	dataframe	Transformed data	Transformed based on fit from perf_PCA

3. classifier class:

For classification, evaluation of results based on several key metrics, plotting of ROC and PRC curves.

Member variables:
data: training data
labels: training labels
tata: test data
tabels : test labels
n = number of data points of training data
nf = number of features of training data
tn = number of data points of test data
tnf = number of features of test data
model: object of classifier selected in perfxxx () methods

Function	Parameters (Defaults included)	Returns	Function
perfGNB	None	None	Perform Gaussian Naïve Bayes. Store output of training and testing predictions.
perfSVM	kernel = 'rbf', C=1, gamma='scale', valflag(deprecated)	None	Perform SVM. Store output of training and testing predictions. Record time taken.
perfMLP	layer sizes = [100,], solver = 'adam', learning rate = 'constant' , learning rate initialization = '0.0001', epochs – 1000	None	Perform MLP. . Store output of training and testing predictions. Record time taken.
perfKNN	neighbours=3	None	Perform KNN. Store output of training and testing predictions. Record time taken.
perfRF	estimators = 10	None	Perform Random Forests. Store output of training and testing

			predictions. Record time taken.
accuracies	True and target training and test labels with an option to externally provide labels to calculate accuracy instead of object members. (extflag = True)	Train and test accuracies	Calculate accuracy for training and testing
confusion_matrix	true and target labels of object or external source (extflag= True)	confusion matrix, numpy array	Compute confusion matrix
c_report	true and target labels of object or external source (extflag= True)	None	display classification report
add_stat	true and target labels of object or external source (extflag= True), verbose	negligence	Display negligence, specificity, sensitivity, false positive rate, positive predicate value if verbose = true
cost	true and target labels of object or external source (extflag= True)	cost	Compute cost of classifier based on given cost equation
get_pred	data, probflag	predictions, probabilities if probflag = True	Get predictions based on fit member variable model of object
draw_roc	test data and labels of object or external source (extflag= True)	probability list, auc, f1	Draw ROC curve
draw_prc	test data and labels of object or external source (extflag= True)	None	Draw Precison/Recall curve